OULUN SEUDUN
AMMATTIKORKEAKOULU

Tuomas Kokko

ETÄOHJATTAVAN LÄMPÖTILASÄÄTIMEN SUUNNITTELU JA TOTEUTUS ARDUINO ETHERNET -MIKROKONTROLLERILLA

ETÄOHJATTAVAN LÄMPÖTILASÄÄTIMEN SUUNNITTELU JA TOTEUTUS ARDUINO ETHERNET -MIKROKONTROLLERILLA

Tuomas Kokko
Opinnäytetyö
Syksy 2012
Automaatiotekniikan koulutusohjelma
Oulun seudun ammattikorkeakoulu

TIIVISTELMÄ

Oulun seudun ammattikorkeakoulu
Automaatiotekniikan koulutusohjelma, Insinööri

Tekijä(t): Tuomas Kokko
Opinnäytetyön nimi: Etäohjattavan lämpötilasäätimen suunnittelu ja toteutus Arduino Ethernet -mikrokontrollerilla
Työn ohjaaja(t): Heikki Takalo-Kippola
Työn valmistumislukukausi ja -vuosi: Syksy 2012
Sivumäärä: 33 + 3 liitettä

Tämän opinnäytetyön aiheena oli suunnitella etäohjattava lämpötilansäädin. Päätavoitteena on etähallintakäyttöliittymän ja laitteiden välisen tiedonsiirron toteutus säätöosan kehittämisen jäädessä toissijaiseksi. Työ toteutetaan Arduino Ethernet -mikrokontrolleria ja Ethernet-lähiverkkoa hyödyntäen. Valmistunutta työtä tulee voida mahdollisesti käyttää pohjana jatkokehitykselle Arduino-ympäristössä. Työn toimeksiantaja toimii Oulun seudun ammattikorkeakoulun, Tekniikan yksikkö.

Ensimmäisessä vaiheessa perehdyttiin lämpötilansäätimiin sekä tämän käyttöpaikkoihin ja hyötyihin. Toisessa vaiheessa käytiin läpi Arduinoon liittyviä tietoja ja faktoja. Tämä käydään läpi jotta saadaan tietoa mikrokontrolleriperheeseen liittyvistä ominaisuuksista, mahdollista jatkokehitystä ajatellen. Kolmannessa vaiheessa tutustuttiin Ethernet-verkon ominaisuuksiin, jotta saadaan taustatietoa työssä käytettävästä tiedonsiirtomenetelmästä. Neljännessä vaiheessa tarkastellaan työn suoritusta sekä testausta.

Lopputuloksena syntyi mobiilisovellus joka kykenee vaihtamaan tietoja Arduino-mikrokontrollerin ja sovelluksen välillä. Sovellus toimii runkona, johon voidaan tarvittaessa lisätä haluttuja ominaisuuksia.

Asiasanat: Arduino, Android, automaatio, lämpötilansäädin

SISÄLLYS

TERMIT JA LYHENTEET

AI	Analoginen tulo
AO	Analoginen lähtö
Arduino	Mikrokontrolleri
DI	Digitaalinen tulo
DO	Digitaalinen lähtö
EEPROM	Haihtumaton puolijohdemuisti
Ethernet	Pakettipohjainen lähiverkkoratkaisu
I/O	Tulo/lähtö
IP-protokolla	Internet-kerroksen protokolla
microSD	SD-korttiperheen pienin korttityyppi
SRAM	Muistityyppi, joka ei vaadi virkistystä mutta vaatii jatkuvan jännitteen
TCP	Kuljetuskerroksen tiedonsiirtoprotokolla
TCP/IP	Tietoverkkoprotokollien yhdistelmä

1 JOHDANTO

Tämän opinnäytetyön on tilannut Oulun seudun ammattikorkeakoulun, tekniikan yksikkö. Työn aiheena on suunnitella Arduino Ethernet -mikrokontrollerilla toteutettu etäohjattava lämpötilansäädin. Päätavoitteena on etähallintakäyttöliittymän ja laitteiden välisten tiedonsiirron toteutus, säätöosan kehittämisen jäädessä toissijaiseksi. Käyttöliittymällä tulee voida lukea lämpötilatiedot Arduinolta sekä muuttaa säätöparametreja. Käyttöliittymän tulee olla helppokäyttöinen, jotta sitä voidaan käyttää ilman aikaisempaa tutustumista pienellä perehdytyksellä. Laitteiden välinen kommunikointi hoidetaan Ethernet-verkon kautta.

Tässä työssä toissijaisena on säätö, joka tehdään ylimääräisen ajan puitteissa. Säätimellä tullaan mittaamaan sekä ulko- että sisälämpötilat. Ulkolämpötilan perusteella ohjataan venttiilimoottoria. Mittaus toteutetaan kahdella NTC-termistorilla, jotka on kytketty Arduino-mikrokontrolleriin. Arduinolle on annettu tietyt säätöparametrit, joiden perusteella määritetään ohjaussignaali.

2 LÄMPÖTILANSÄÄTIMET

Lämpötilansäätimien yksi sovelluspaikka on kiinteistöautomaatiossa, jossa säädin voi säätää esimerkiksi lämminvesi- tai patteriverkoston lämpötilaa. Säätimiä käytetään myös teollisuudessa ja erilaisissa laitteissa, joihin tarvitaan aktiivista lämpötilan tarkkailua ja säätöä. Teollisuudessa on usein tarvetta pitää laitteiden lämpötilat sopivina. Näin voidaan ennalta ehkäistä liiallisesta lämmöstä johtuvaa kulumista ja ylläpidetään hyvä hyötysuhde koko prosessin ajan. Myös useat teollisuudenprosessit vaativat tarkan lämpötilansäädön, jotta saavutetaan haluttu lopputulos.

Kiinteistön lämpötilansäädin mittaa joko pelkästään ulkolämpötilan tai ulko- ja sisälämpötilat. Säädin laskee mittausten perusteella ohjausarvot asetettujen/ohjelmoitujen algoritmien avulla. Ohjaussignaali lähetetään toimilaitteelle, säätimen laskettua ohjausarvot. Ohjattavana toimilaitteena voi toimia esimerkiksi lämminvesiverkostoon tarkoitettu moottoriventtiili. Lämmönlähteenä voi toimia öljypoltin, pellettipoltin, ilmalämpöpumppu, sähkölämmittimet, aurinko jne.

Useimpiin nykyään rakennettaviin kiinteistöihin asennetaan jonkinlainen elektroninen lämpötilasäädin. Säädin voi olla yksittäinen säädin tai sellainen joka on sulautettuna erilliseen kiinteistöautomaatiojärjestelmään. Kiinteistöön asennettavalla lämpötilansäätimellä voidaan tavoittaa huomattavia kustannussäästöjä, kun pidetään tilat halutussa lämpötilassa. Sekä yli- että alilämmittämiseltä säästytään lämpötilan ollessa tarkasti säädetty. Näin saavutetaan käyttömukavuutta pienemmän lämpötilanvaihtelun vuoksi ja samalla energiankulutus vähenee.

Jotta lämpötilasäätimen hankinta koettaisiin mielekkäänä, tulisi sen täyttää tiettyjä kriteereitä, kuten

- alhaiset aloitus- ja käyttökustannukset
- kuoletettava hankintahintansa
- varmatoimisuus
- helppo asennettavuus
- laajennettavuus
- helppokäyttöisyys.

3 ARDUINO

Arduino on vapaaseen lähdekoodiin perustuva kehitysalusta, jota voidaan käyttää monenlaisiin projekteihin. Arduinoa voidaan käyttää tietokoneen yhteydessä, tuoden anturitiedot tietokoneen käsiteltäviksi. Sitä voidaan käyttää myös omana yksikkönään käsitellen tuloihin tulevat tiedot oman ohjelman mukaisesti. Arduino ja sen ohjelmisto toimii Windows, Macintosh OSX ja Linux -käyttöjärjestelmissä, kun taas useimmat mikrokontrollerit toimivat vain Windows-ympäristössä.

Arduino-ohjelmointiympäristö perustuu processing-ohjelmointiympäristöön ja käytettävä ohjelmointikieli perustuu Wiring-ohjelmointikieleen. Arduinon yksinkertainen ja selkeä ohjelmointiympäristö/ohjelmointikieli on helppokäyttöinen aloittelijoille. Se on kuitenkin sen verran teknisesti kehittynyt, jotta sitä voidaan käyttää vaativimmissakin projekteissa. (1.)

3.1 Historia

Arduinon kehitys alkoi projektina, Massimo Banzin ja David Cuartiellesin toimesta vuonna 2005, Interaction Design Instituutissa Ivreassa, Italiassa. Banzin ja Cuartiellesin tarkoitus oli tehdä kehitysalusta, joka oli kylliksi halpa opiskelijoille. Alustan tuli myös olla teknisesti modernimpi kuin vuonna 2005 oli yleisesti saatavilla. Projektista tehtiin vapaaseen lähdekoodiin perustuva, joten jokaisella on mahdollisuus osallistua kehitykseen tai kehittää itse omaa ohjelmisto Arduinon lähdekoodin perusteella. (3.)

Arduino on saavuttanut paljon suosiota helppokäyttöisyytensä vuoksi. Yli 300000 Arduinoa oli myyty toukokuussa 2011. Suosion yhtenä syynä on varmasti Arduinojen hinnat, jotka ovat suhteellisen edullisia. Hinnat asettuvat 20–80€:n välille riippuen mallista (2).

3.2 Alustat

Arduinon valikoimissa on useita erilaisia alustoja, jotka sopivat erilaisiin tarkoituksiin. Ominaisuudet vaihtelevat, niin tulojen/lähtöjen, muistien määrien kuin muiden ominaisuuksien osalta. Muita ominaisuuksia ovat mm. muistikorttien liitettävyys, Ethernet, muut erilaiset kommunikointi tavat, jne.

Alla olevissa taulukoissa on esitetty alustojen tietoja. Taulukossa 1 on esitetty eri alustojen tulojen ja lähtöjen määriä. Taulukossa 2 esitetään prosessoreiden tietoja ja muistien määriä. Käytettävä alusta valitaan yleensä näiden ominaisuuksien perusteella.

TAULUKKO 1. Arduinojen I/O ominaisuuksia

	Digital I/O	PWM	Analog Input	Analog Output
Uno	14	6	6	0
Leonardo	20	7	12	0
Mega2560	54	14	16	0
Due	54	12	12	2
Ethernet	14	4	6	0
Mega ADK	54	15	16	0
Micro	20	7	12	0
Nano	14	6	8	0
Fio	14	6	8	0
Mini	14	6	8	0
Pro Mini	14	6	6	0
LilyPad	14	6	6	0
Pro	14	6	6	0

TAULUKKO 2. Eri Arduino-alustojen prosessoreihin ja muisteihin liittyvää tietoa

	Cpu	Alter. Cpu	Clock speed (MHz)	Bit	Voltage (V)	Flash (kB)	SRAM (kB)	EEPROM (kB)
Uno	ATmega 328	n/a	16	8	5	32	2	1
Leonardo	ATmega32u4	n/a	16	8	5	32	2,5	1
Mega2560	ATmega2560	n/a	16	8	5	256	8	4
Due	Cortex-M3	n/a	84	32	3,3	512	96	0
Ethernet	ATmega328	n/a	16	8	5	32	2	1
Mega ADK	ATmega2560	n/a	16	8	5	256	8	4
Micro	ATmega32u4	n/a	16	8	5	32	2,5	1
Nano	ATmega328	n/a	16	8	5	32	2	1
Fio	ATmega328P	n/a	8	8	3,3	32	2	1
Mini	ATmega328	n/a	16	8	5	32	2	1
Pro Mini	ATmega168	n/a	(8)/16	8	(3,3)/5	16	1	0,5
LilyPad	ATmega168V	ATmega328V	(8)/16	8	(3,3)/5	16	1	0,5
Pro	ATmega168	ATmega328	(8)/16	8	(3,3)/5	(16)/32	(1)/2	(0,5)/1

3.3 Shieldit

Arduinoihin on saatavana lisäosina erilaisia shieldejä, joilla saadaan lisää ominaisuuksia alustoihin.

Arduino Motor Shield

Motor Shield perustuu L298-piiriin, joka kykenee ohjamaan induktiivisia kuormia, kuten releitä, solenoideja, tasasähkö- ja stepperi-moottoreita.

Arduino Proto Shield

Proto Shied on kehitysalusta, johon asennetaan halutut komponentit kolvaamalla.

CAN-BUS Shield

CAN-BUS Shield antaa Arduinolle kyvyn liittyä CAN-väylään, jota käytetään ajoneuvotekniikassa paljon.

CAN-BUS Shieldin ominaisuudet:

- CAN v2.0B 1MB/s
- nopea SPI-liitäntä
- CAN-yhteys sub-D-liittimellä
- microSD-ominaisuus
- liitäntä LCD:lle.

Ethernet Shield

Ethernet Shieldillä saadaan Arduino-alustaan Ethernet- ja microSD-ominaisuus. Shield perustuu Wiznet W5100 -Ethernet-piiriin, jonka ominaisuuksiin kuuluu:

- 10/100Mb:n tiedonsiirtonopeus
- RJ-45-liitäntä
- tuki neljälle yhdenaikaiselle yhteydelle
- tuki Server tai Client -moodeille.

Wifi Shield

Wifi Shieldillä saadaan Arduino-alustaan langaton yhteys. Shield perustuu HDG104 Wireless LAN 802.11b/g -piiriin. Tämän shieldin ominaisuuksiin kuuluu

- tuki 802.11b/g-verkkoihin
- tuki WEP- ja WPA2-salauksiin
- microSD-korttipaikka

Wireless SD Shield

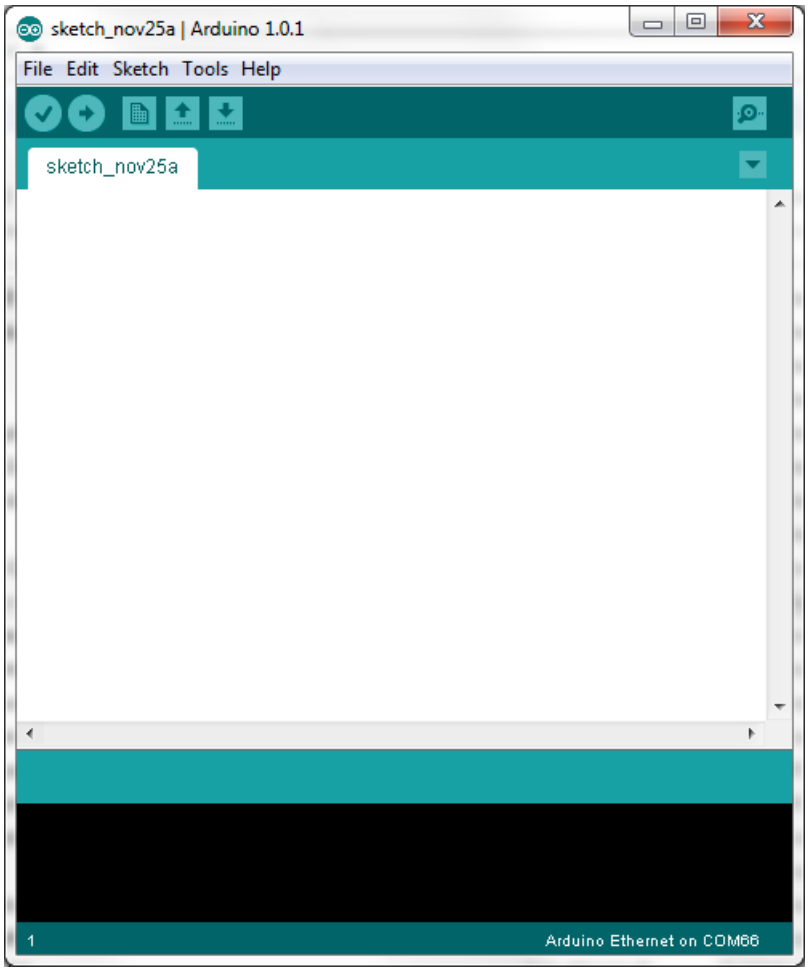
Wireless SD Shieldiin voi asentaa moduulin (esim Xbee), joka mahdollistaa langattoman käytön. Shieldissä on myös microSD-ominaisuus.

Wireless Proto Shield

Wireless Proto Shield on kehitysalusta, johon asennetaan halutut komponentit kolvaamalla. Tätä shield:iä on mahdollisuus käyttää langattomasti esim. Xbee-moduulilla.

3.4 Arduino-ohjelmisto

Arduinon ohjelmoimiseen voi käyttää Arduinon omaa ohjelmistoa, joka perustuu Processing-ohjelmointiympäristöön. Ohjelmiston saa ladattua ilmaiseksi Arduinon omilta kotisivuilta. Arduino-ohjelmisto toimii Windows, Macintosh OSX ja Linux -käyttöjärjestelmissä, kun taas useimmat mikrokontrollerien ohjelmoimistyökalut toimivat vain Windows-ympäristössä. (1.)



KUVA 1. Arduino-ohjelmisto, versio 1.0.1

4 ETHERNET

Ethernet on käytetyin lähiverkkotekniikka useiden etujensa vuoksi. Ethernet on halpa, nopea, luotettava ja helppokäyttöinen. Lisäksi Ethernet-laitteiden saatavuus on hyvä. Näiden seikkojen takia kiinnostus Ethernet-verkkotekniikkaan on suurta. Ethernet-tekniikka on kasvavissa määrin käytössä teollisuus- ja kiinteistöautomaatiossa. (11.)

Fyysisesti perinteinen Ethernet perustuu väylätopologian käyttöön, eli kaikki verkossa olevat koneet on kytketty samaan kaapeliin. Yleisin nykyään käytetty topologiarakenne on tähtitopologia. Tähtitopologian etuja ovat mm. joustava toteutettavuus ja topologian sallima suuri nopeus. Muita verkkotopologioita on rengas-, väylä- ja puutopologiat. (11.)

Ethernet käyttää tiedonkuljetukseen TCP/IP-protokollaa.

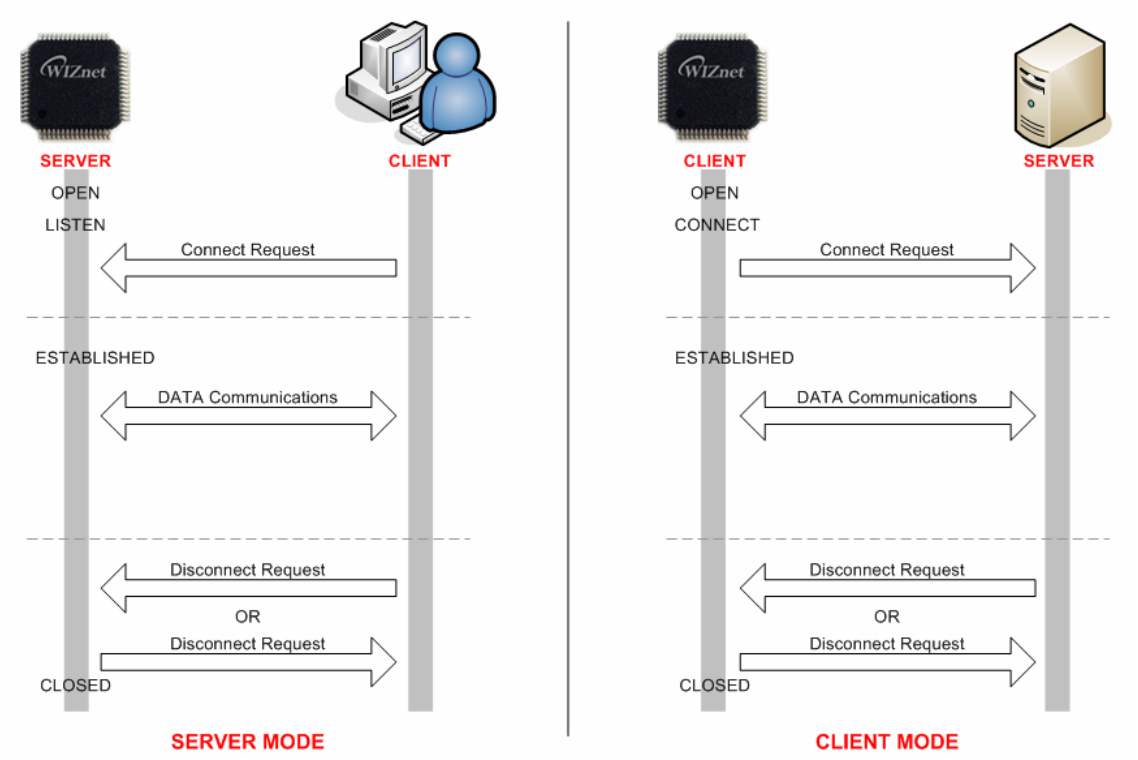
4.1 TCP

TCP (Transimission Control Protocol) on tietoverkkoprotokolla, jolla muodostetaan yhteyksiä laitteiden välille. Tieto lähetetään verkossa IP-protokollaa hyväksi käyttäen.

TCP-yhteys muodostetaan kolmitiekättelyllä. Asiakas lähettää SYN-paketin palvelimelle, johon palvelin vastaa lähettämällä SYN ACK-paketin. Asiakas lähettää ACK-paketin saatuaan SYN ACK-paketin palvelimelta. Tämän kättelyn jälkeen voi alkaa tiedonsiirto. Tiedonsiirron onnistuminen varmistetaan monella eri mekanismilla.

Yhteys päätetään nelitiekättelyllä. Asiakas lähettää FIN-paketin palvelimelle. Palvelin vastaa lähettämällä ensin ACK-paketin ja tämän jälkeen FIN-paketin. Asiakas lähettää ACK-paketin palvelimelle.

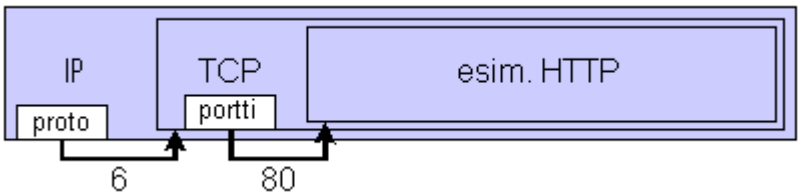
Kuvassa 2 on esitetty yksinkertaistettu yhteyden muodostus, tiedonlähetys ja yhteyden lopetus. Käytettävässä W5100-Ethernet-piirissä on mahdollisuus valita joko palvelin- tai asiakastila.



KUVA 2. TCP-yhteyden muodostus (4)

4.2 IP-protokolla

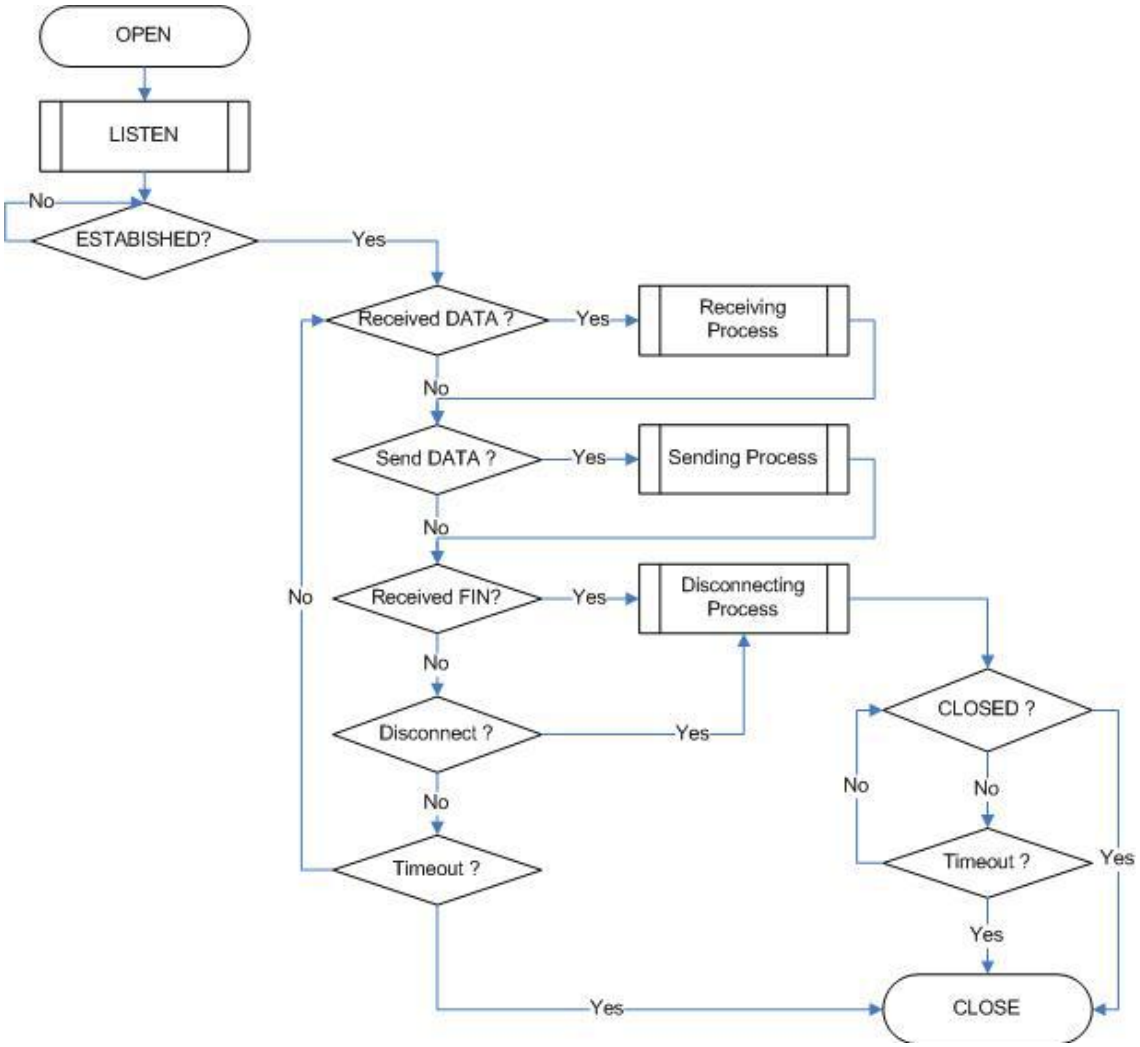
IP on TCP/IP-mallin verkkokerroksen protokolla, joka vastaa IP-tietoliikennepakettien toimittamisesta Internet-verkossa. IP-paketissa kuljetettavat protokollat on numeroitu. Protokollan numerosta vastaanottaja tietää, mitä paketin sisällä on (Kuva 3). (12.)



KUVA 3. IP-paketti (12)

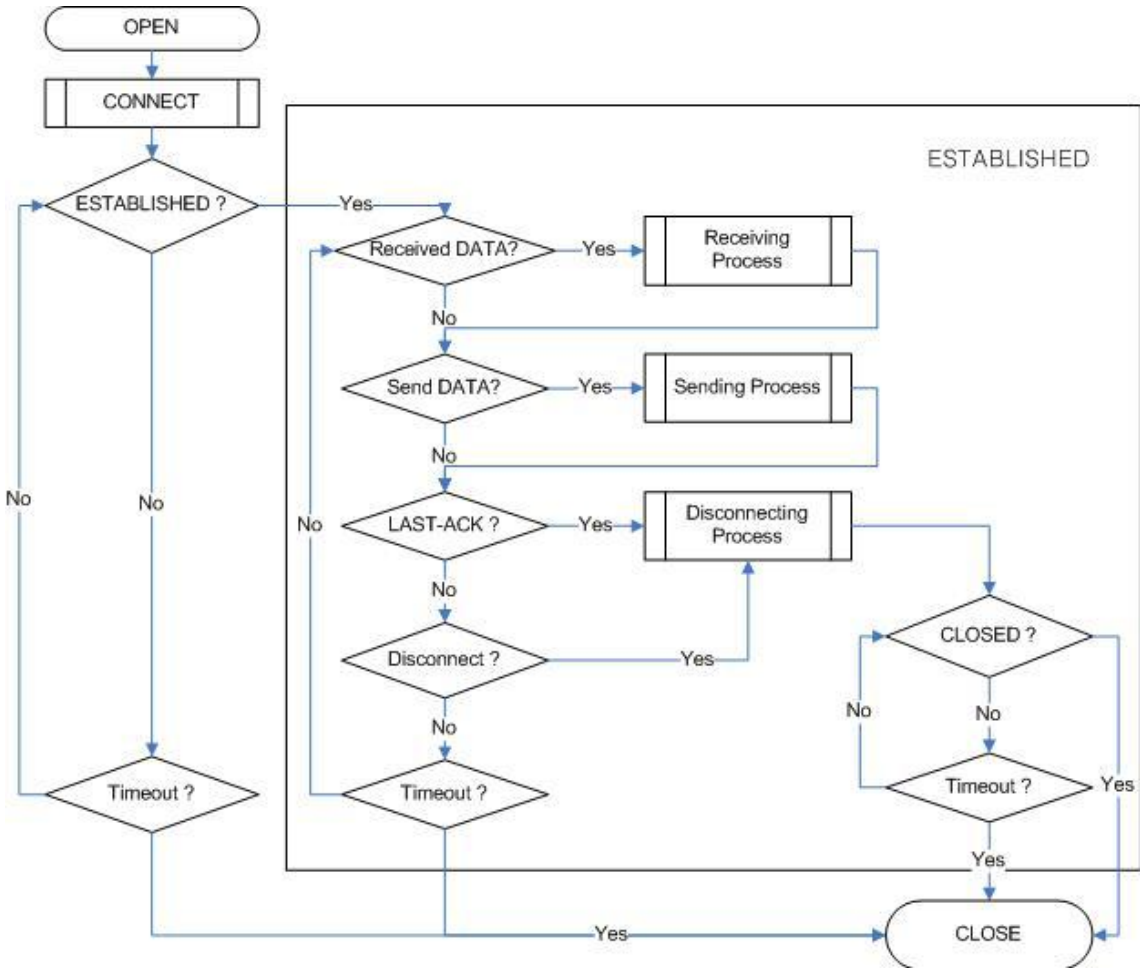
4.3 Palvelin/Asiakas

W5100-Ethernet-piirissä on mahdollisuus valita joko palvelin- tai asiakastila. Palvelintilassa laite on kokoajan avoin tuleville yhteyspyynnöille. Pyynnön tullessa palvelin toimii kuvan 4 sekvenssin mukaisesti.



KUVA 4. Palvelintilan yhteydenluonti sekvenssi (4)

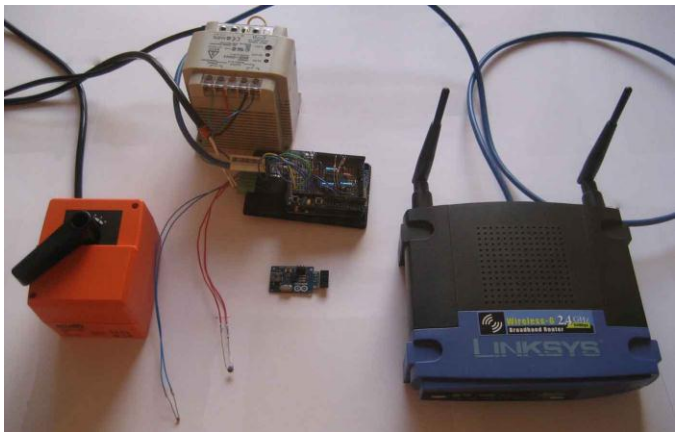
Asiakastilassa yhteys palvelimeen otetaan vain tarvitessa. Palvelimelta saatavat palvelut voivat olla esimerkiksi tietojen varastoimista, prosessointia tai tietojen välittämistä. Yhteys palvelimelle otetaan kuvan 5 sekvenssin mukaisesti.



KUVA 5. Asiakastilan yhteydenluonti sekvenssi (4)

5 KOKOONPANO JA OHJELMOINTI

Työ alkoi tutustumalla saatujen laitteistojen ominaisuuksiin ja tutkimalla miten ne saadaan toimimaan yhdessä. Eniten aikaa kului Arduino-mikrokontrollerin parissa, koska se oli kohtalaisen tuntematon laite. Laitteiden kytkentä toisiinsa oli helpohkoa koska laitteet ovat fyysisesti suhteellisen yksinkertaisia (kuva 6).



KUVA 6. Käytettävä laitteisto

5.1 Laitteisto

Tätä työtä aloitettaessa oli käytössä seuraava laitteisto:

- 24 voltin jännitelähde
- Arduino Ethernet
- Arduino Prototype Shield v.5
- Belimo HTC24-SR
- Huawei U8800
- Linksys WRT54G
- tietokone

- USB/Serial Light Adapter.

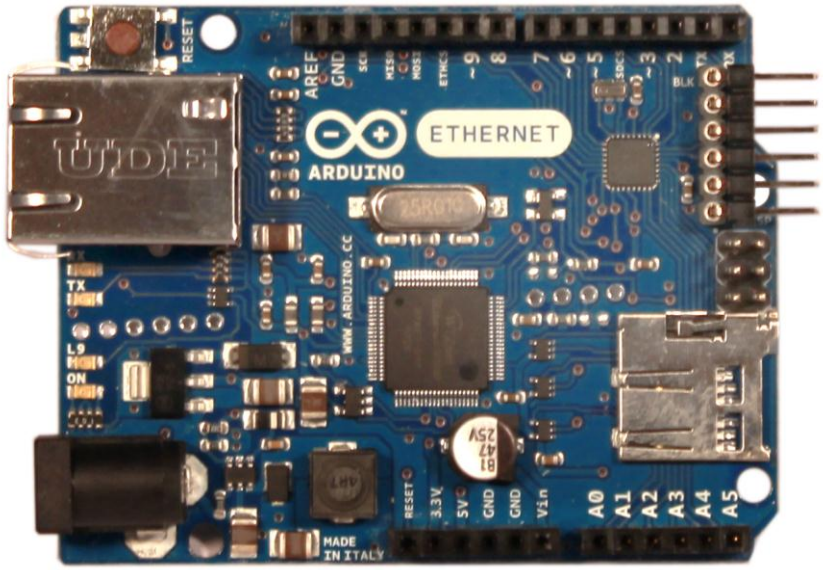
5.1.1 Arduino Ethernet

Keskeisenä komponenttina on Arduino Ethernet -mikrokontrolleri (kuva 7). Arduino Ethernet on ATmega328-mikropiiriin perustuva mikrokontrolleri. ATmega328 sisältää 32 kB flash-muistia (joista on 0,5 kB käynnistysohjelman käytössä), 2 kB SRAM-muistia ja 1 kB EEPROM-muistia.

Arduino Ethernetissä on 14 digitaalista pinniä, joista 4:ää voidaan käyttää PWM lähtöinä. PWM-lähtö on 8-bittinen, joka tarkoittaa 256 erilaista lähtöjännitetasoa. Digitaalipinneistä 10,11, 12 ja 13 ovat varattuja SPI:lle ja pinni numero 4 on varattu SD-kortille. Näitä ei suositella muuhun käyttöön, jolloin käyttöön jää 9 pinniä.

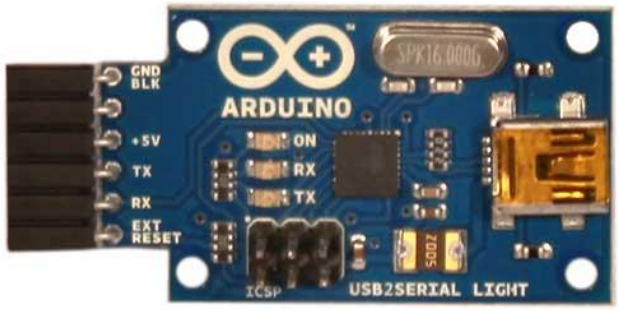
Analogisia tuloja on 6 kappaletta. Tulot kykenevät 10-bitin resoluutioon, eli kortti kykenee havaitsemaan jännitemuutoksen noin 5 mV:n tarkuudella.

Erikoisominaisuutena kortissa on Ethernet-ominaisuus sekä microSD-korttipaikka. Ethernet-ominaisuus on toteutettu W5100 TCP/IP-Ethernet-kontrollerilla. W5100:n ominaisuuksiin kuuluu tuki seuraaville TCP/IP-protokollille: TCP, UDP, ICMP, IPv4 ARP, IGMP, PPPoE. Ethernet-kontrollerin ansiosta Arduino Ethernettiä voidaan käyttää sekä palvelimena (Server) että asiakkaana (Client). Liitäntänä Ethernet-verkkoon toimii standardi RJ-45-kaapeli.



KUVA 7. Arduino Ethernet -mikrokontrolleri (1)

Ohjelmointi tapahtuu erillisellä tietokoneeseen kytketyllä USB-sarja-adapterilla tai FTDI/USB-kaapelilla. Tässä työssä käytetään Arduinon omaa USB/Serial Light Adapteria (kuva 8).



KUVA 8. USB Serial Light Adapter (1)

5.1.2 Arduino Prototype Shield v.5

Tässä työssä tarvitaan Arduino Ethernetin päälle kytkettävä Prototype Shield, johon on kolvattu operaatiovahvistin (LM324N). Operaatiovahvistinta tarvitaan, koska Arduinon maksimissaan 5 voltin ulostulojännite tulee saada nostettua 10 volttiin. 10 voltin jännite on tarpeen, koska ohjattavana laitteena tässä työssä on HTC24-SR-venttiilimoottori. Venttiilimoottorin ohjausjännite on 0–10V.

5.1.3 Belimo HTC24-SR

HTC24-SR-venttiilimoottori on suunniteltu lämmitysjärjestelmien tarpeisiin. Venttiilimoottorinkaapelissa on neljä johdinta. Punaiseen ja mustaan johtoon tuodaan 24 voltin käyttöjännite. Numero 3:lla merkittyyn valkoiseen johtoon tuodaan ohjausjännite 0–10V. Numero 5:lla merkitystä johdosta saadaan asentotieto. Venttiilimoottorissa on käsi- ja automaattikäyttömahdollisuus.

Tärkeimmät tiedot ovat seuraavat:

- vääntömomentti: 5 Nm
- ajoaika 70s
- ohjaustapa 0–10 VDC
- käyttöjännite 24 VDC.
- käsi/automaattikäyttö

5.1.4 Huawei U8000

Arduinoa varten suunnitellun käyttöliittymän testilaitteena toimi Huawei U8800 -älypuhelin. U8800 on varustettu Androidin 4.0.4-käyttöjärjestelmällä ja tuella 802.11b/g/n-standardin mukaiseen WLAN-lähiverkkotekniikkaan. Ominaisuuksiensa perusteella U8800 on sopiva testiväline tähän työhön.

5.1.5 Linksys WRT54G

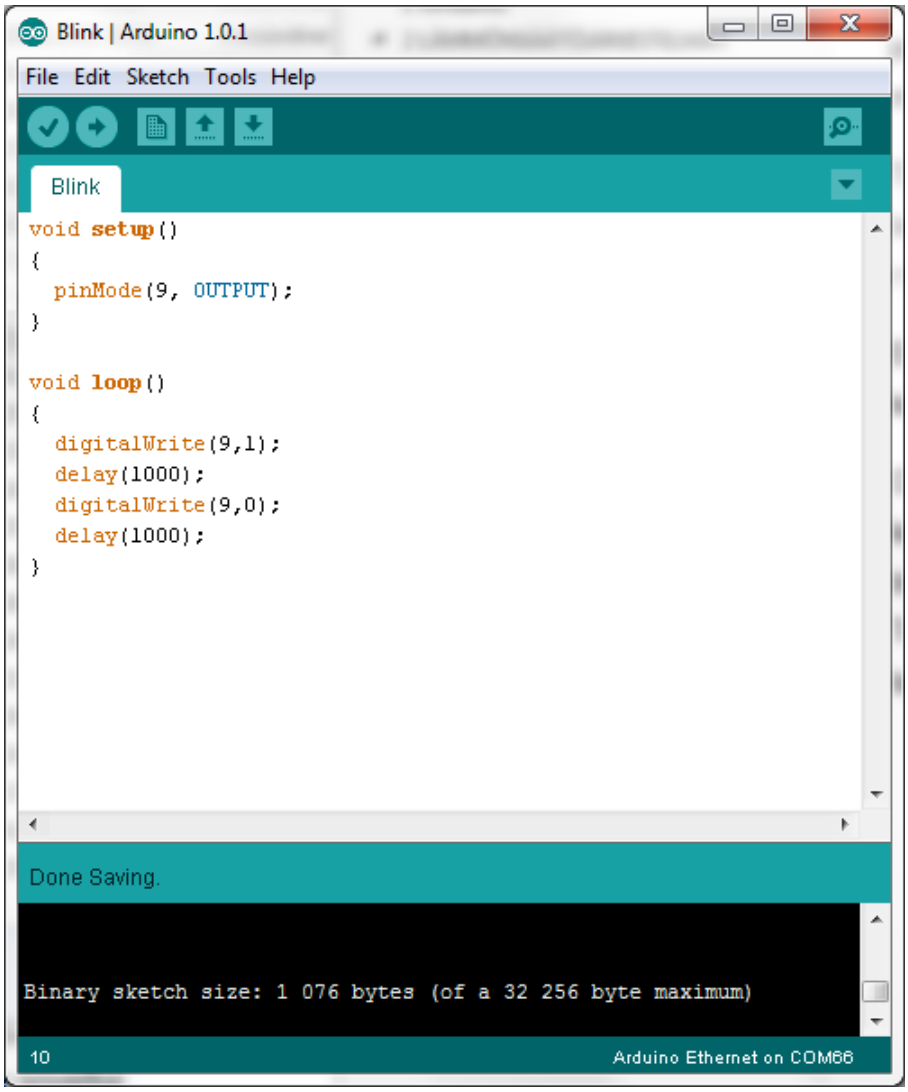
WRT54G-langaton reititin on tietoverkkoja yhdistävä laite. Tähän laitteeseen Arduino Ethernet yhdistetään RJ-45-kaapelilla, jolloin reititin antaa Arduinolle

IP-osoitteen. Myös mobiililaite saa IP-osoitteensa reitittimeltä langatonta yhteyttä käyttäen. Reititin toimii tietoja välittävänä linkkinä Arduinon ja mobiililaitteen välillä.

5.2 Arduinon ohjelmointi

Arduinon ohjelmointi alkoi harjoitussovelluksia tutkimalla. Harjoituksia tutkimalla ohjelmointikieli tuli perusominaisuuksiltaan nopeasti tutuksi. Ohjelmointikieli on suhteellisen helposti ymmärrettävä, sen helpoksi tehtyjen ominaisuuksiensa vuoksi. Ohjelman runko koostuu vähimmillään setup- ja loop-funktioista. Setup-funktiossa määritellään kerran ajettavat komennot, kuten tulojen ja lähtöjen määrittäminen, sarjayhteyden muodostaminen jne. Loop-funktioon kirjoitetaan itse ohjelma, jota toistetaan loputtomasti. Ohjelman kasvaessa tulee myös usein tarve tehdä itse uusia funktioita. Näin pidetään ohjelma helposti luettavana.

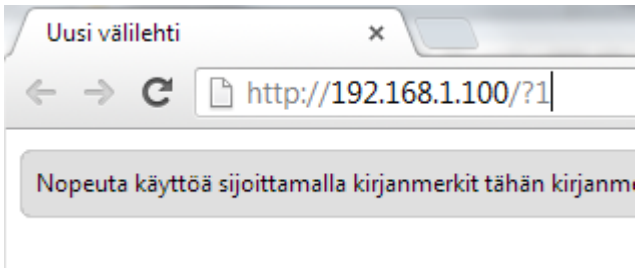
Ensimmäisenä perusohjelmointiharjoituksen tuloksena syntyi ohjelmanpätkä. Ohjelma toimi siten että sisäänrakennettu LED digitaaliportissa yhdeksän laitettiin syttymään ja sammumaan sekunnin välein. (Kuva 7.)



KUVA 9. Arduinon ohjelmointitesti

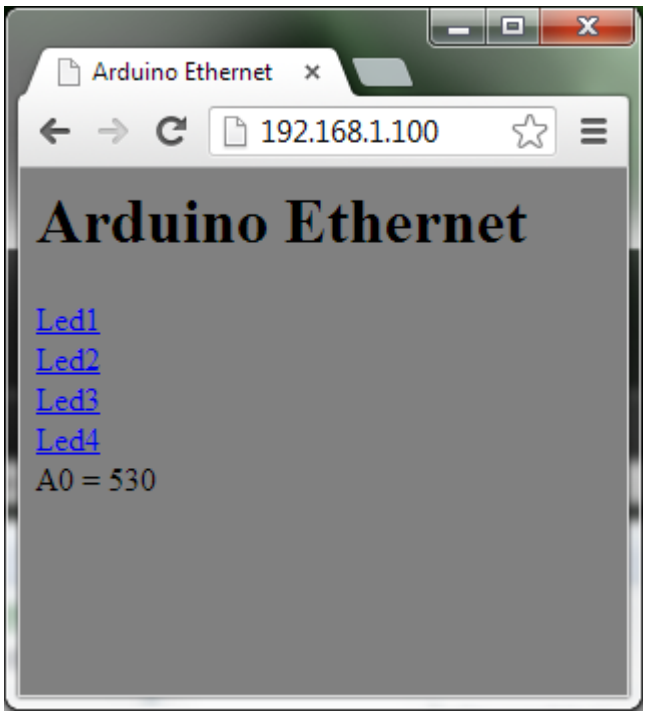
Perustestien jälkeen aloitettiin kokeilut Ethernetin käyttämisestä Arduinossa. Esimerkki Arduino-palvelimen luomisesta löytyy Arduinon kotisivuilta. Koodi löytyy Liitteestä 2.

Testit alkoivat komennon lähettämisestä lähiverkon kautta. Samassa lähiverkossa olevan laitteen verkkoselaimen hakuvalikkoon kirjoitettiin ledin sytyttävä komento. Komento koostuu kahdesta osasta, Arduinon IP-osoitteesta ”192.168.1.100/” ja komennosta, jonka Arduino tulkitsee ”?1” (kuva 10).



KUVA 10. LEDin sytyttävä käsky, joka on kirjoitettuna verkkoselaimeen

Testin onnistuttua kirjoitettiin testi HTML-sivu, jonka Arduino lähettää pyytävälle asiakkaalle. Aikaisempaa harjoitusta hyödyntäen HTML-sivulle tuli neljä linkkiä, joista saadaan neljä eri LEDiä syttymään. Arduino kirjoittaa sivulle myös analogisen tulon arvon. (Kuva 9.)



KUVA 11. HTML-käyttöliittymätesti

Tästä harjoituksesta oli tarkoitus tulla pohja, jonka perusteella lopullinen käyttöliittymä rakennetaan. Käyttöliittymän etuja oli yksinkertaisuus ja se että sitä pystytään käyttämään jokaisella laitteella, jossa on verkkoselain.

Ottaen huomioon Arduinon rajoitteellisen muistikapasiteetin johtopäätökseksi tuli ettei kannata sulauttaa HTML-koodia Arduinon ohjelmakoodiin. Päätöksen teon jälkeen ongelmaksi tuli kuinka toteuttaa sovellus. Vaihtoehtoina oli yrittää käyttää Arduino Ethernetin omaa microSD-ominaisuutta ja tallentaa verkkosivut microSD-kortille. Tällöin Arduino olisi lukenut HTML-sivun microSD-kortilta ja lähettänyt sen verkkoselaimelle.

Toisena vaihtoehtona oli ulkopuolisen sovelluksen tekeminen. Ensimmäisen vaihtoehdon haittapuolena oli se, että Arduino pystyy lukemaan vain yhtä tiedostoa kerrallaan. Yhden tiedoston rajoitus oli esteenä tehdä visuaalisesti miellyttävä käyttöliittymä, mahdollisimman vähillä uhrauksilla. microSD-korttia voi myös tarvita myöhemmin johonkin muuhun tarkoitukseen (kuten datalogger-käyttöä ajatellen).

Tavoitteena oli myös käyttää mahdollisimman vähän Arduinon resursseja, ajatellen virrankulutustakin. Lopuksi päädyttiin ulkopuoliseen sovellukseen joka noutaa muuttujat Arduinolta.

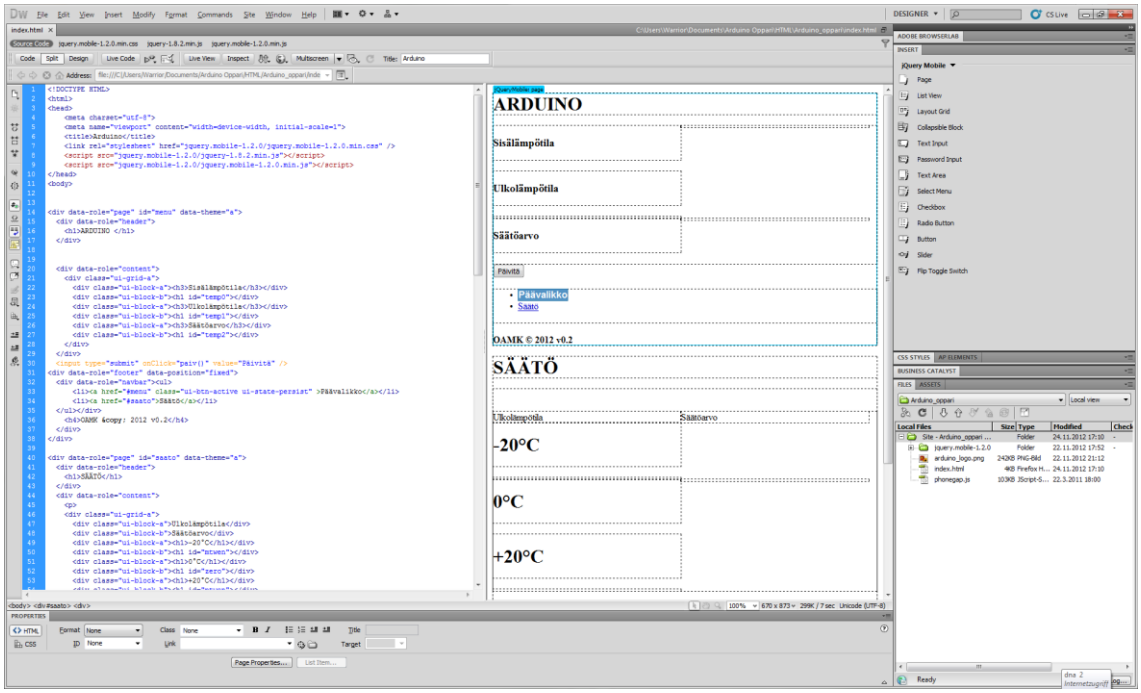
5.3 Sovelluksen toteuttaminen

Sovelluksen toteutus alkoi tutustumisella eri toteutusvaihtoehtoihin. Mobiilisovelluksen tulisi tukea Android-ympäristöä, koska testilaitteena oli Android-pohjainen mobiililaite. Androidin sovellukset tehdään Java-ohjelmointikielellä, jonka opettelemiseen rajoitetun aikamäärän puitteissa ei ollut mahdollisuuksia.

Toisena vaihtoehtona oli jQuery Mobile. jQuery Mobile on tarkoitettu mobiiliverkkosivujen ja sovellusten tekemiseen. jQuery Mobile ei ole ohjelmointikieli vaan ohjelmistokehys, joka muodostaa rungon rakennettavan sivuston päälle. Sen käyttäminen sopi tähän projektiin helpon käytettävyyden ja helposti saavutettavan, hyvän visuaalisen ulkonäön vuoksi. (9)

Sivusto kirjoitetaan HTML-ohjelmointikielellä (tai kuvauskielellä), johon lisätään jQuery Mobile -scripti. Valmiista sivuista on mahdollista tehdä mobiilisovellus, jonka voi asentaa älypuhelimelle.

Käyttöliittymän tekemisessä käytettiin Adobe Dreamweaver -ohjelmistoa, joka osoittautui käytettävyydeltään hyväksi työkaluksi (kuva 12).



KUVA 12. Adobe Dreamweaver -ohjelmisto

Käyttöliittymän ollessa testausvaiheessa, ongelmaksi tuli Arduinolta saatavan datan yhdistäminen mobiilisovellukseen. Ongelmana olivat tietoliikenneverkkojen rajoitukset. Verkkoselaimiin on asetettu rajoitus siitä kuinka vapaasti saadaan hakea tietoa toiselta domainilta. Monien yleisimpien tiedonvälityskielien testauksen jälkeen, ongelmaksi tuli aina sama rajoitus, eli Same Origin Policy (7). Same Origin Policy:llä estetään tietojen haku eri verkko-osoitteesta, kuin missä hakeva sivu on. Kyseinen esto on tehty turvallisuussyistä.

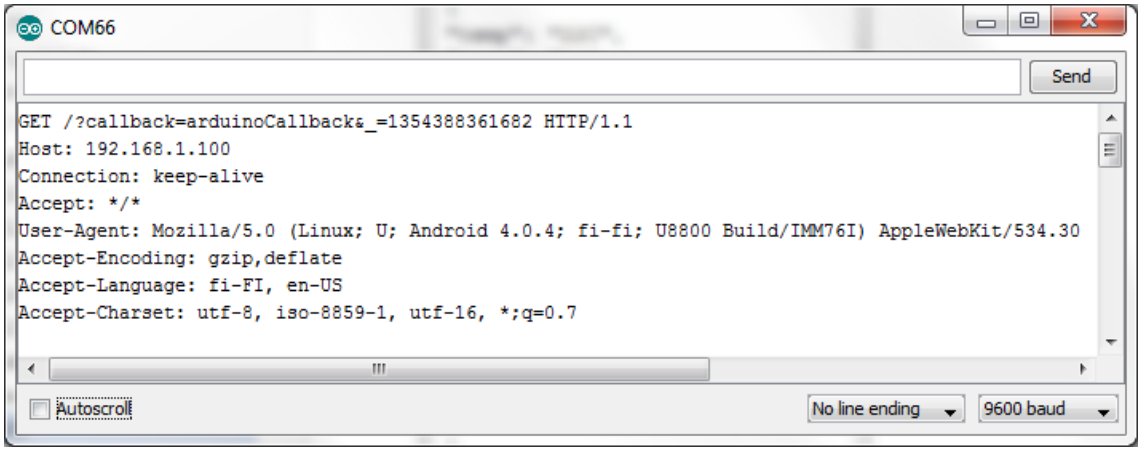
Ratkaisuksi kyseiseen estoon löytyi JSONP (JavaScript Object Notation with Padding) -tiedonsiirtomenetelmä. JSONP sallii tiedon haun eri verkko-osoitteesta, kiertäen Same Origin Policy -rajoituksen.

Koska JSONP kiertää Same Origin Policy:n, voi esiin nousta tietoturvaongelmia. Ongelmia ilmenee, jos pyydetään epäilyttävältä palvelulta JSONP-menetelmällä tietoja. Palvelu voisi mahdollisesti lähettää haitallisen JacaScript-koodin. Lähetettävä koodi voi olla esimerkiksi sellainen, joka kalastelisi käyttä-

jän henkilökohtaisia tietoja ja lähettäisi tiedot eteenpäin. Ongelmaa ei synny tässä työssä, koska lähettäjä ja vastaanottaja ovat yhden henkilön hallinnassa. Lisäturvaa tuo myös se, että Arduino ja mobiililaite ovat suojatussa lähiverkossa.

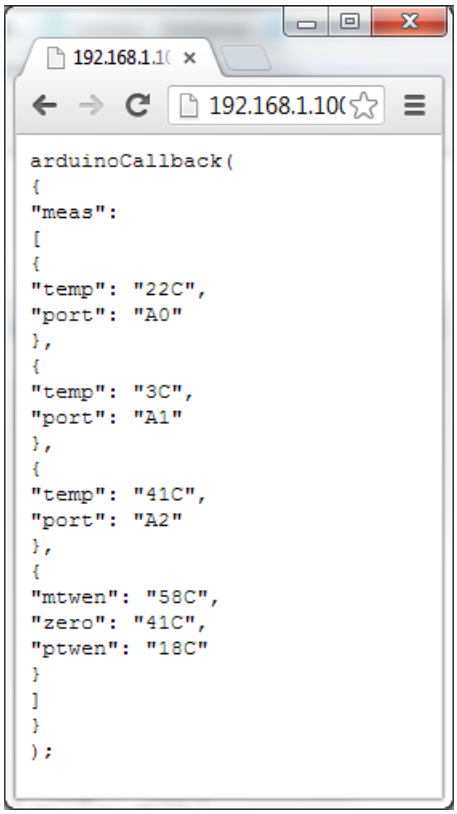
6 TESTAUS

Testaus alkaa mobiilisovelluksen avaamisella. Avattaessa sovellus lähettää viestin Arduinolle. Viestiä voi tarkastella Arduinon-ohjelman Serial Monitorin ylimmältä riviltä. Viesti käyttää GET-menetelmää arduinoCallback-funktion kutsuun. (Kuva 13.)



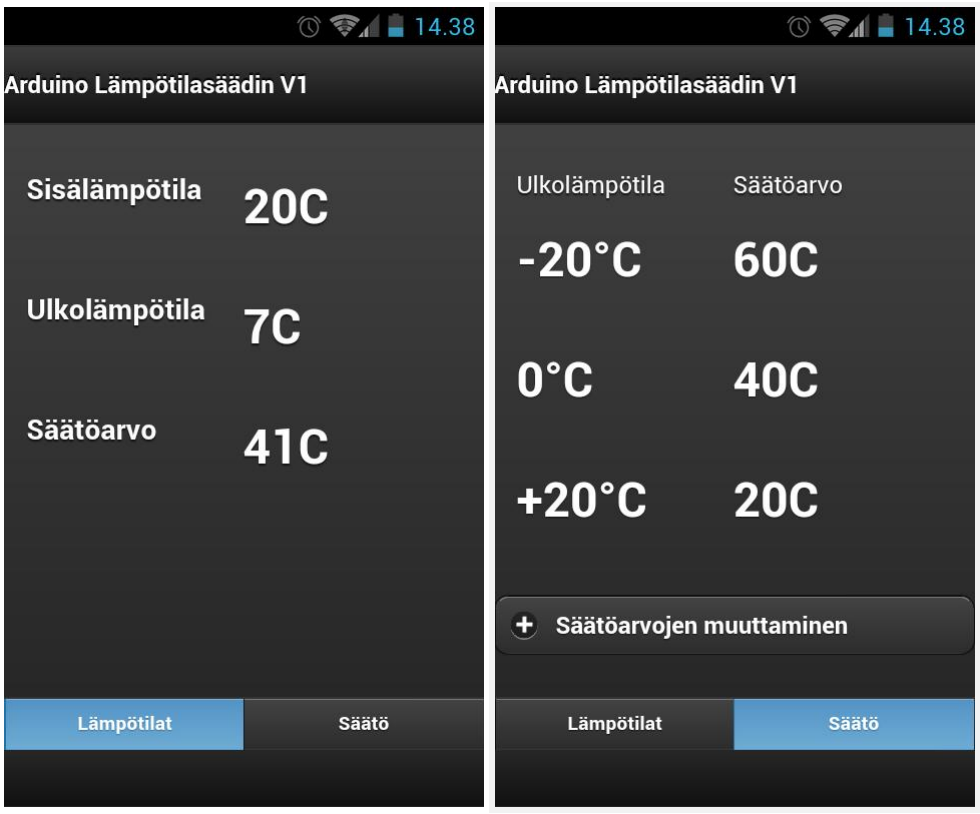
KUVA 13. Arduinon saama viesti Serial Monitorissa

Arduinon saadessa viestin lähetetään takaisin JSONP-viesti. Viestissä lähetetään kaikki tiedot jotka on tarkoitus saada Android-sovellukselle. Testausvaiheessa viestissä lähetetään satunnaisia lukuja, eikä oikeita mittaustuloksia. Satunnaisluvut lähetetään, koska tässä vaiheessa laitteistoa ei ole kytketty toisiinsa mutta halutaan varmistaa tiedonsiirron toimivuus. Viesti koostuu funktion nimestä ja objekteista (kuva 14). JSONP-menetelmän vuoksi viestissä olevia arvoja voidaan helposti tarkastella ilman että sitä tarvitaan dekoodata. Viestissä tullaan lähettämään ulko- ja sisälämpötilat ja sen hetkinen säätölämpötila patteriverkoston lämpötilalle. Patteriverkoston lämpötilaa ei tässä työssä varmisteta takaisinkytkennällä, vaikkakin sisälämpötilan anturin voisi tähän tehtävään valjastaa. Viestissä olevat mtwen, zero ja ptwen ovat säätöarvoja, joiden perusteella tullaan määrittelemään ohjausarvo.



KUVA 14. Arduinon lähettämä testi JSONP-viesti

Arduinon lähettämän viestin tiedot näytetään sovelluksen lämpötila- ja säätönäytössä (Kuva 15). Sovellus hakee päivitetyt arvot Arduinolta tietyn ajan välein. Päivitysnopeus määräytyy sovellettavasta kohteesta. Esimerkiksi kiinteistökäytössä päivitysnopeus voi olla monia minuutteja lämpötilanmuutoksen hitauden vuoksi.



KUVA 15. Mobiilisovelluksen lämpötila- ja säätöikkunat.

Säätöikkunassa on mahdollisuus säätöarvojen muuttamiseen. Liukusäätimet tulevat esiin painettaessa ”Säätöarvojen muuttaminen” -painiketta. Liukusäätimillä voidaan valita haluttu säätöarvo, minkä jälkeen nämä arvot lähetetään painamalla Lähetä Arduinolle -painiketta. Arduino käyttää hyväksi näitä tietoja ohjausarvoa määritellessään. (Kuva 16.)

KUVA 15. Mobiilisovelluksen säätöarvojen muuttaminen

7 POHDINTA

Tämän opinnäytetyön tarkoituksena oli etäohjattava lämpötilansäädin suunnittelu ja toteutus. Päätavoite oli etähallintakäyttöliittymän ja laitteiden välisten tiedonsiirron toteutus. Ethernet-verkkoa käytettiin laitteiden välistä kommunikointia varten. Säätimen tarkoitus oli mitata ulko- sekä sisälämpötila. Ulkolämpötilan perusteella lasketaan ohjausarvo jolla ohjataan moottoriventtiiliä.

Ensisijaisena tavoitteena oli saada tehtyä käyttöliittymä, jota voidaan ohjata lähiverkossa mobiililaitteella tai vaihtoehtoisesti tietokoneella. Työlle asetetut päätavoitteet saavutettiin mielestäni hyvin käyttöliittymän ja laitteiden kommunikoinnin osalta. Käyttöliittymän toteuttaminen mobiililaitteelle onnistui hyvin ja tarvittava tieto saadaan lähetettyä langattomassa lähiverkossa. Langattomuus ominaisuudesta saavutetaan etuja jos säädin on vaikeasti pääsyisessä paikassa tai säädin on paikassa jossa on vaadittu tiivistä kotelointia kuten kosteissa paikoissa.

Käyttöliittymästä tuli helppokäyttöinen, yksinkertainen, toimiva ja se on helposti muutettavissa eri käyttötarkoituksia varten. Kommunikointi laitteiden kanssa saatiin onnistumaan, vaikkakin Arduinon ja mobiilisovelluksen välisen tiedonsiirron ongelmien kanssa kului liian suuri osa opinnäytetyön tekemisen ajasta. Tästä syystä jouduttiin joistain ideoista luopumaan.

Työhön jäi vielä paljon paranneltavaa ja ideoita jäi toteutumatta. Tiukan aikataulun vuoksi esimerkiksi lämmönsäätimen säätötapa jäi puutteelliseksi. Säätö oli aluksi tarkoitus toteuttaa syöttämällä kolmea lämpötilaa vastaavat säätölämpötilat. Tämän jälkeen ohjelma laskee kulmakertoimen ja tätä vastaavan ohjausarvon. Työn lopuksi säätö toimi kolmiportaisesti minkä vuoksi ei päästä ideaaliseen lämpötilan tarkkuuteen. Suunnitelmissa oli myös säädön toteuttaminen PID-säätöalgoritmeja käyttäen.

Vaikka tässä työssä tehty lämpötilasäädin on suunniteltu kiinteistökäyttöä silmälläpitäen, sitä voidaan käyttää myös moniin muihin käyttökohteisiin pienin muutoksin.

Jatkokehitystä ajatellen käyttöön voisi ottaa microSD-ominaisuuden ja ohjelmoida Arduinon tallentamaan tietoja taulukkoon, joka tallennetaan microSD-kortille aikatietojen kanssa. Taulukon tiedot voidaan lähettää pyydettäessä etälaitteelle. Muidenkin toimilaitteiden lisääminen ei olisi mahdoton ajatus, vaikka käytetyssä Arduino Ethernetissä on I/O:ta erittäin rajoitetusti.

Etäyhteyksiäkin olisi hyvä tarkastella syvällisemmin, kuten voiko Arduinon tietoturvallisesti liittää Internet-verkkoon jolloin etäyhteys voisi onnistua vaikka toiselta puolelta maailmaa. Tämä mahdollistaa esimerkiksi mökin lämmittämisen, jotta se olisi lämpimänä sinne saavuttaessa. Arduinon monikäyttöisyyden vuoksi siihen voisi kehittää myös vaikka varashälyttimen joka lähettää viestin hätätilanteessa.

Ominaisuuksien laajentamista varten suositeltavissa olisi Arduino Megan ja Ethernet- tai Wi-Fi-shiedin ostaminen. Näillä saavutetaan etuja, koska niissä on enemmän liitäntöjä.

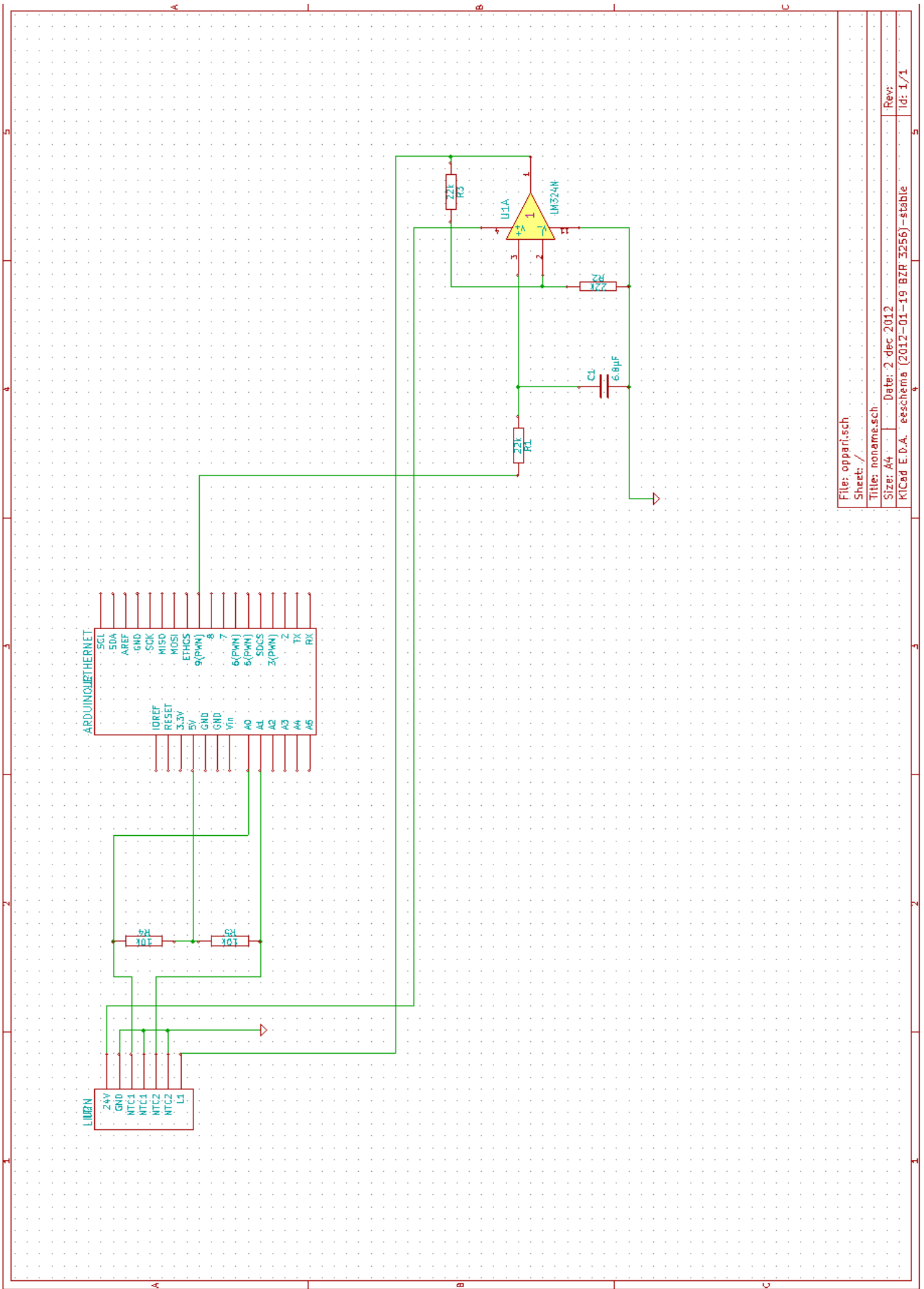
Työn laaja aihe opetti paljon, johtuen suuresta aihepiiristä joita piti työn mahdollistamiksi opetella. Näitä taitoja on hyvä hyödyntää ja kehittää tulevaisuudessa.

LÄHTEET

1. Arduino. Saatavissa: http://www.arduino.cc/. Hakupäivä 10.10.2012
2. Paeae.com. Saatavissa: http://paeae.com/. Hakupäivä 20.11.2012
3. Arduino The Documentary. Saatavissa: http://arduinothedocumentary.org/. Hakupäivä 19.11.2012
4. W5100 Datasheet. Versio1.1.6. 2008. WIZnet Co. Saatavissa: http://www.sparkfun.com/datasheets/DevTools/Arduino/W5100_Datasheet_v1_1_6.pdf. Hakupäivä 21.11.2012
5. Belimo HTC24-SR. Saatavissa: http://www.belimo.fi/pdf/SaneerausNR-sarja.pdf. Hakupäivä 21.11.20.12
6. JSON-P: Safer cross-domain Ajax with JSON-P. Saatavissa: http://www.json-p.org/. Hakupäivä 29.11.2012
7. Same Origin Policy. Saatavissa: http://www.w3.org/Security/wiki/Same_Origin_Policy/. Hakupäivä 29.11.2012
8. Developer Tools. Saatavilla: http://developer.android.com/tools/index.html. Hakupäivä 29.11.2012
9. jQuery Mobile. Saatavissa: http://jquerymobile.com/. Hakupäivä 29.11.2012
10. w3schools. Saatavissa: http://www.w3schools.com/. Hakupäivä 29.11.2012
11. Häyrinen, Tuomas. IHA-3300/ELE-3350 Mekatroniikan erityiskysymyksiä -tutkielma. Saatavissa: http://www.ele.tut.fi/teaching/ele-3350/ethernet.pdf. Hakupäivä 29.11.2012
12. Wikipedia, Vapaa tietosanakirja 2012. Saatavissa: http://fi.wikipedia.org/wiki/IP. Hakupäivä 1.12.2012.

```
/*
  Web Server

 A simple web server that shows the value of the analog input pins.
 using an Arduino Wiznet Ethernet shield.

 Circuit:
 * Ethernet shield attached to pins 10, 11, 12, 13
 * Analog inputs attached to pins A0 through A5 (optional)

 created 18 Dec 2009
 by David A. Mellis
 modified 9 Apr 2012
 by Tom Igoe

 */

#include <SPI.h>
#include <Ethernet.h>

// Enter a MAC address and IP address for your controller below.
// The IP address will be dependent on your local network:
byte mac[] = {
  0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress ip(192,168,1, 177);

// Initialize the Ethernet server library
// with the IP address and port you want to use
// (port 80 is default for HTTP):
EthernetServer server(80);

void setup() {
 // Open serial communications and wait for port to open:
  Serial.begin(9600);
   while (!Serial) {
    ; // wait for serial port to connect. Needed for Leonardo only
  }


  // start the Ethernet connection and the server:
  Ethernet.begin(mac, ip);
  server.begin();
  Serial.print("server is at ");
  Serial.println(Ethernet.localIP());
}


void loop() {
  // listen for incoming clients
  EthernetClient client = server.available();
```

```
 if (client) {
  Serial.println("new client");
  // an http request ends with a blank line
  boolean currentLineIsBlank = true;
  while (client.connected()) {
   if (client.available()) {
    char c = client.read();
    Serial.write(c);
    // if you've gotten to the end of the line (received a newline
    // character) and the line is blank, the http request has ended,
    // so you can send a reply
    if (c == '\n' && currentLineIsBlank) {
     // send a standard http response header
     client.println("HTTP/1.1 200 OK");
     client.println("Content-Type: text/html");
     client.println("Connnection: close");
     client.println();
     client.println("<!DOCTYPE HTML>");
     client.println("<html>");
          // add a meta refresh tag, so the browser pulls again every 5 seconds:
     client.println("<meta http-equiv=\"refresh\" content=\"5\">");
     // output the value of each analog input pin
     for (int analogChannel = 0; analogChannel < 6; analogChannel++) {
      int sensorReading = analogRead(analogChannel);
      client.print("analog input ");
      client.print(analogChannel);
      client.print(" is ");
      client.print(sensorReading);
      client.println("<br />");
     }
     client.println("</html>");
     break;
    }
    if (c == '\n') {
     // you're starting a new line
     currentLineIsBlank = true;
    }
    else if (c != '\r') {
     // you've gotten a character on the current line
     currentLineIsBlank = false;
    }
   }
  }
  // give the web browser time to receive the data
  delay(1);
  // close the connection:
  client.stop();
  Serial.println("client disonnected");
 }
}
```


22k
R3
U1A
LM324N
22k
R2
C1
6.8µF
22k
R1
ARDUINOWETHERNET
SCL
SDA
AREF
GND
SCK
MISO
MOSI
ETHCS
9(PWM)
8
7
6(PWM)
5(PWM)
SDCS
3(PWM)
2
TX
RX
IOREF
RESET
3.3V
5V
GND
GND
Vin
A0
A1
A2
A3
A4
A5
10k
R4
10k
R5
LUOZN
24V
GND
NTC1
NTC1
NTC2
NTC2
L1
File: oppari.sch
Sheet: /
Title: noname.sch
Size: A4
Date: 2 dec 2012
Rev:
KiCad E.D.A. eeschema (2012-01-19 BZR 3256)-stable
Id: 1/1